**RESUMEN Y COMENTARIO DE CÓDIGO LIMPIO**

**Apellidos y Nombres** Israel Mallma Perez

**Doctorado en Sistemas**

**Semestre I**

**Asignatura** Tópicos avanzados de ingeniería de software

* **RESUMEN Y COMENTARIO DE CÓDIGO LIMPIO**

El autor afirma que: si el código no está limpio, puede poner de rodillas a una organización de desarrollo. Cada año, se pierden incontables horas y recursos significativos debido a un código mal escrito. Pero no tiene por qué ser así.

El experto en software Robert C. Martin presenta un paradigma revolucionario con Codigo Limpio, que ha ayudado a llevar los principios ágiles desde el punto de vista de un profesional a decenas de miles de programadores, se ha asociado con sus colegas; para destilar su mejor práctica ágil de limpieza de código, en un libro que inculcará los valores del artesano del software y te convertirán en un mejor programador, pero solo si trabajas en ello.

Este libro es imprescindible para cualquier desarrollador, ingeniero de software, gerente de proyecto, líder de equipo o analista de sistemas interesado en producir un mejor código. Debemos considerarnos autores que asumamos la responsabilidad de comunicar bien nuestro código a los lectores. La relación entre el tiempo dedicado a la lectura y la escritura del código es de 10:1, por lo que la lectura del código debería facilitarse incluso si dificulta la escritura.

* **RESUMEN Y COMENTARIO DEL CAPÍTULO I:**

El autor Robert C. Martin afirma que: escribir código limpio es lo que debes hacer para llamarte profesional y no hay excusa razonable para hacer algo menos que lo mejor posible

A continuación se presentan algunos de los puntos como:

* La atención a los detalles es la base fundamental de la profesionalidad más que cualquier gran visión.
* El 80 por ciento del trabajo en software es mantenimiento. Una onza de prevención vale una libra de cura. El software debe escribirse centrándose en el mantenimiento y no en la producción.
* Hacer que tu código sea legible es tan importante como hacerlo ejecutable
* Aplaude la cosmovisión japonesa que comprende el valor crucial del trabajo diario constante y aplica ese punto de vista en el desarrollo de software para enfatizar el valor de la simplicidad del código, el cuidado diario y la consistencia.
* Señala que la sabiduría danesa que aconseja la honestidad en las cosas pequeñas es tan importante como prestar atención a las cosas pequeñas. Nos aconseja que seamos honestos con el código, con nuestros colegas sobre el estado del código y, lo que es más importante, que seamos honestos con nosotros mismos sobre nuestro código. Afirma que ni la arquitectura ni el código limpio insisten en la perfección, solo en la honestidad y en hacer lo mejor que podamos.
* **RESUMEN Y COMENTARIO DEL CAPÍTULO 2: NOMBRES SIN SENTIDO**

En esta parte del texto nos indica, qué los códigos tienen una funcionalidad insubsistencia, además nos habla de la complejidad de la función que está contiene en las diferentes programaciones, resalta el porque no solo es la simplicidad del código sino el carácter implícito, los nombres debe de ser claros. Los resaltantes de esto es que los nombres de bebés suelen ser más adecuados qué los nombres extensos, siempre y cuando sean claros.

El nombre de una variable, función o clase debe responder una serie de cuestiones básicas. Debe indicar por qué existe, qué hace y cómo se usa. Si un nombre requiere un comentario, significa que no revela su cometido. El nombre no revela nada. No evoca una sensación de tiempo transcurrido, ni de días. Debe elegir un nombre que especifique lo que se mide y la unidad de dicha medida. Lo más complicado a la hora de elegir un buen nombre es que requiere habilidad descriptiva y acervo cultural.